A Framework for Business Mashup Applications

Vijay K. Naik

IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA
vkn@us.ibm.com

Abstract. Mashup applications for business can lead to better business insights, marketing opportunities, and may provide opportunity to monetize the value locked in business data. To enable necessary experimentation with mashups by domain experts, we describe a development and runtime framework that exposes data level abstractions in the form of templates and automates the programming details associated with plumbing for mashups. The framework also provides runtime support to manage security, reliability, and other business application related enterprise IT concerns.

Keywords: Business mashup applications · Cloud service orchestration

1 Introduction

Mashups are web based applications that typically combine data, content, and application functions from multiple services on the internet and present those in a meaningful cohesive manner. Mashups may filter out subject specific types of information, blend different types of data to create new insights and even new information. By combining data from different sources, mashups help users uncover and understand hidden patterns, recognize correlations, gain competitive advantage, create business value, or simply create pure entertainment value. Data sources tend to be either static, or slowly changing, or dynamic and fast changing. GPS data, geographical data, national/state/city boundaries, webcam locations, language constructs, societal values etc. are examples of static or slowly changing data. Weather data, twitter trends, news headlines, stock prices, traffic patterns, webcam feeds, consumer spending patterns, public opinions, fashion trends, spread of deceases, demographics, census data are all examples of dynamically changing data. Combining data that change at different frequencies can often lead to interesting insights, bring out correlations, or help in making predictions in one dimension using information from another dimension. By juxtaposition of fast and slow changing and static forms of information and presenting in a contextually clear manner, mashups can attract more users, provide insights, and create value meaningful to the users.

Mashup applications provide an attractive opportunity for businesses to conduct market analysis, brand management, customer relations, forecasting, and for monetizing the value locked in enterprise data. However, to fully realize this potential value, domain experts need to experiment with many different combinations and patterns. In addition, the resulting application must also address security, reliability, and other enterprise IT concerns. Today this requires domain expertise as well as low level programming skills to manage integration with web services and IT management services. This is often a high barrier for domain experts and for software engineers. Existing frameworks for mashup applications do not address many of the requirements inherent to business and enterprise environments.

In this paper, we describe a mashup development and run-time environment that (i) enables users to develop mashup applications without requiring deep software engineering or programming skills, (ii) provides composition mechanisms for encapsulating IT management functions with the mashup runtime and automates common integration patterns.

2 Mashup Business Applications

Mashup applications for business combine business sensitive data with external publically available and/or third party commercial data. These applications may be used internally for business analysis purposes or may be provided externally to customers, partners, and other interested parties. Like other business applications, mashup applications for business also need to satisfy certain non-functional requirements including security and privacy requirements adhering to compliance and regulatory concerns, availability and performance requirements, and failure handling according to predefined SLAs. To address these concerns, enterprise IT typically follow best practices for deploying and managing applications and associated supporting middleware and infrastructure services. The same practices need to be followed for mashup applications for businesses.

Enterprise IT mandate special handling of both business sensitive data and third party data: access to business sensitive data must be properly protected to remain compliant with various regulations while allowing access to the key aspects via the mashup applications; and externally obtained data needs to properly filtered and fenced off from the rest of the enterprise environment to meet security and compliance requirements. When external applications access enterprise data, proper care must be taken to address firewall issues to allow applications and services to access only the authorized data by authorized external parties without compromising the security of the rest of the enterprise.

Because mashups open up possibilities for combining data in numerous different ways and utility of a mashup application may be short lived and possibly of interest only to a small audience, mashup development processes need to be fast requiring low IT or development effort. Even end users who are not professional software engineers should be able to construct a mashup application with relatively minimum amount of effort and investment of time.

3 A Framework for Mashup Business Applications

A viable framework for developing and managing mashup business applications must provide: (i) low barrier to entry for domain experts and other users with low programming skills for developing mashup applications using only their domain knowledge; (ii) address for security, reliability, and other IT concerns; and (iii) provide enterprise grade support for mashup applications.

Our framework is based on the concept of using domain specific templates for capturing and providing the integration logic specific to each domain. Examples of domain specific templates are templates for Customer relations for brand management, Customer service for order tracking, Customer service for competitive positioning, Sales positioning for competitive pricing, Decision support for product placement, Transportation logistics for same day delivery, Risk modeling using real-time data, Emergency response for natural and man-made disasters, and so on. The templates are extensible and mechanisms are provided to import new templates. Templates incorporate domain specific data sources -- both public and private. From the template, script based composition tools are synthesized which then guide the end user to create their own mashup application within the context of that template. The composition tools provide access to functions for performing filtering, aggregation and other statistical operations, and analytics such as map-reduce on the data received from the data sources both pre- and post-data-mashup. Templates encapsulate the details of accessing data sources as well as manage the data flow to and from data operators. The mashup developer can focus on the data abstractions and on the content without having to worry about the underlying plumbing. In case of complex mashup applications, mashup developers may want to perform complex custom analysis during the mashup or post-mashup steps. This can be done by incorporating the mashup application within a server-side application developed using scripting languages such as Ruby or Node.js. Such an application would then provide the necessary logic for post-mashup analysis. Once a mashup application is developed using a template, it can then be previewed and published as a web application to be used by others.

Templates are extensible. However, extending the templates require some degree of programming experience and knowledge about the APIs supported by specific data provider web services. Similarly new domain specific templates can be added to the framework. Again this requires some degree of software engineering skills.

A library of data operators is provided to manipulate various types of structured and unstructured data streams, content feeds, and feeds from other mashup applications. This library is extensible and new operators can be added to the library. The operators can be in the form of URLs to services, self-contained applications, or modules to existing packages. Template developers may enable a subset of data operators that are suitable for a particular step in the flow which are then made available to mashup developer in the form of a catalog that they can use to select and apply for that stage. More experienced users can import their own data operators while developing a mashup application.

Once the mashup application is developed, internally it is represented as BPM workflow which orchestrates the steps of accessing and initializing data sources and executes the flow. Application developers submit their applications along with the specifications of their service requirements. These requirements are expressed in the form of attributes of services delivered by cloud-based infrastructure and platform service providers. The application may also be associated with explicit policies such as secure connections to data-sources, firewall rules, network authentication rules and keys, and the prescribed service levels. The underlying framework prepares the environment by acquiring instances of the necessary IT services from cloud service providers and then deploys and manages the application according to enterprise policies.

4 Related Work

In [1], authors characterize five popular mashup development tools and identify common and unique characteristics among these tools. They also identify desired characteristics that tools and frameworks need to provide for efficient mashup development. In [2], authors provide a survey of cloud service composition approaches and classify these approaches according eight categories they have identified. In [3], authors propose using domain specific mashup language developed using domain specific concepts. Using the language a domain-specific mashup tool and runtime platform is automatically generated. We share their goal of easing the burden of mashup development by using domain specific knowledge at the cost of reduced generality. However our approaches of using domain-specific knowledge differ. In [4], the author describes Microsoft SharePoint as a platform for enterprise mashups.

5 Conclusion

We have outlined a template based framework for developing mashup applications taking into enterprise IT considerations. Framework described here enables application developers to describe the data-sources and organize mashup operations within the context of a template. In case of complex mashup applications users also can specify the flow and operations during various stages of the flow. At run-time, the underlying integration logic and other plumbing needed to enable secure connectivity, enablement of IT management services for application lifecycle operations such scaling, failure handling, backup/recovery are automatically generated from the template. In a future publication we will describe details of template design and their implementation using specific example domains.

References

1. Yu, J., Benatallah, B., Casati, F., Daniel, F.: Understanding Mashup Development. IEEE Internet Computing **12**(5), 44–52 (2008)
2. Jula, A., Sundararajan, E., Othman, Z.: Cloud computing service composition: A systematic literature review. Expert Systems with Applications **41**, 3809–3824 (2014)
3. Soi, S., Daniel, F., Casati, F.: Conceptual development of custom, domain-specific mashup platforms. ACM Trans. Web **8**(3) (2014)
4. Arredondo, J.: SharePoint: A platform for enterprise mashups (2008). http://www.microsoft.com/mashups